BIO Web of Conferences **163**, 01004 (2025) https://doi.org/10.1051/bioconf/202516301004
ICBBB 2025

Graph-based method for constructing consensus trees

Elio Torquet[1,*], *Jesper* Jansson[2,**], and *Nadia* Tahiri[1,***]

[1]Department of Computer Science, University of Sherbrooke, 2500, boul. de l'Université, Sherbrooke, J1K 2R1, QC, Canada
[2]Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, 606-8501, Japan

Abstract. A consensus tree is a phylogenetic tree that synthesizes a given collection of phylogenetic trees, all of which share the same leaf labels but may have different topologies, typically obtained through bootstrapping. Our research focuses on creating a consensus tree from a collection of phylogenetic trees, each detailed with branch-length data. We integrate branch lengths into the consensus to encapsulate the progression rate of genetic mutations. However, traditional consensus trees, such as the strict consensus tree, primarily focus on the topological structure of these trees, often neglecting the informative value of branch lengths. This oversight disregards a crucial aspect of evolutionary study and highlights a notable gap in traditional phylogenetic approaches. In this paper, we extend *PrimConsTree*, an graph-based method for constructing consensus trees. This algorithm incorporates topological information, edge frequency, clade frequency, and branch length to construct a more robust and comprehensive consensus tree. Our adaptation of the well-known Prim algorithm efficiently identifies the maximum frequency branch and maximum frequency nodes to build the optimal consensus tree. This strategy was pre-processed with clustering steps to calibrate the robustness and accuracy of the consensus tree.
Availability and implementation: The source code of PrimConsTree is freely available on GitHub at https://github.com/tahiri-lab/PrimConsTree.

1 Introduction

The exploration of phylogenetic relationships is crucial in biology to reveal evolutionary connections among species. A consensus tree is a computational method used to synthesize a set of phylogenetic trees, aiming to distill the most frequently occurring characteristics into a single representative tree. This approach facilitates the identification of common evolutionary relationships and patterns across multiple phylogenetic analyses [1]. In this context, the accuracy of a consensus tree lies in its ability to effectively represent the collective input while accounting for uncertainty or divergence in the data. However, consolidating multiple trees into a single structure presents a significant challenge [2].

*e-mail: Elio.Torquet@USherbrooke.ca
**e-mail: jj@i.kyoto-u.ac.jp
***e-mail: Nadia.Tahiri@USherbrooke.ca

The difficulty in this process is reconciling variations and conflicts that may arise from a set of phylogenetic trees. Integrating diverse evolutionary perspectives and resolving inconsistencies becomes intricate due to structural differences between trees, often making them incompatible. This mismatch requires careful consideration and the application of advanced computational techniques to construct a coherent and accurate composite consensus tree. In addressing this issue, a proposed solution recommends employing multiple consensus trees for a more comprehensive approach [3, 4].

The phylogenetic tree involves three main elements: 1) topology, 2) branch length, and 3) label position. In phylogenetic trees, topology refers to the branching structure that represents evolutionary relationships among species. It is important to note that closely related species based on topology may not necessarily exhibit morphological similarity. For instance, crocodiles are more closely related to birds than to lizards based on their evolutionary lineage, despite crocodiles and lizards appearing more morphologically similar. This occurs due to differing rates of evolutionary change, particularly rapid morphological evolution along the bird lineage. Branch lengths in phylogenetic trees represent the amount of genetic divergence, commonly measured in nucleotide substitutions. While branch lengths can sometimes be interpreted as indicative of time, this interpretation is contingent upon the assumption of a molecular clock. In cases where terminal branches descend from the same common ancestor, differences in branch lengths are generally interpreted as variations in evolutionary rates rather than direct representations of temporal duration. For instance, a longer branch may reflect a higher rate of substitution rather than a longer period of time when compared to a shorter branch arising from the same ancestral node. The position of the labels, affixed to the ends of the branches, provides essential taxonomic information, revealing evolutionary links through their relative positions.

A crucial element in the accurate depiction of phylogenetic trees is the incorporation of branch lengths, representing evolutionary time or genetic change among species or sequences. These lengths contribute significantly to the understanding of the temporal aspects of evolution, offering insights into the processes shaping the *Tree of Life*. Branch lengths play multiple roles across diverse domains, from assessing phylogenetic diversity to identifying and analyzing selection processes [5–10]. Despite substantial advancements in reconstructing phylogenetic trees, the challenge of constructing consensus trees with more topological information and meaningful branch lengths remains an active research area.

In response to this problem, our study introduces an efficient methodology for constructing Maximum Spanning Trees (MST) that consider edge and clade frequencies. This unified framework aims to integrate information from diverse phylogenetic inference methods and data sources, culminating in a consensus tree encapsulating widely supported branches and their associated branch lengths. Our approach promises a refined and comprehensive perspective on evolutionary relationships, shedding light on the consensus time scale of evolution.

2 Definitions and notation

This article uses standard *phylogenetic tree* terminology. A phylogenetic tree is a rooted, directed tree where every internal node has at least two children and each leaf has a distinct label. Given a tree T, the set of all nodes in T is $V(T)$ and it includes two disjoint subsets: the leaf nodes $L(T)$, and the internal nodes (including the root node) $I(T)$. The set of edges in T is $E(T)$ and an edge $(u, v) \in E(T)$ denotes a directed link from node u to v. An edge (u, v) represents a parent-child relation where u is the parent of v and v is the child of u. The length of an edge (u, v) in T is denoted by $dist_T(u, v)$ and represents the amount of genetic change from u to v. Given a node $u \in V(T)$, $T[u]$ means the subtree of T rooted at u.

In a phylogenetic tree, a *clade* is anny subset of the leaves that have a common ancestor such that no other leaves in the tree have that same node as an ancestor. Given a tree T, each node $u \in V(T)$ represents a distinct clade that is defined as $C_u = L(T[u])$. A clade C_u is said to *be supported* by T if there exists a $v \in V(T)$ with $L(T[v]) = C_u$. Two distinct clades C_u and C_v are said to be *compatible* either if $C_u \subseteq C_v$, $C_v \subseteq C_u$ or $C_u \cap C_v = \emptyset$.

The set of input trees $S = \{T_1, \ldots, T_k\}$ is a set of k phylogenetic trees, all sharing the same set of leaves $L(S) = L(T_1) = \cdots = L(T_k)$. The parameters used to measure the size of the input are $k = |S|$ for the number of trees and $n = |L(S)|$ for the number of leaves, respectively. We refer to the subset of trees in S that support the clade C_u by S_u, i.e., for every internal node we define $S_u = \{T \in S : u \in V(T)\}$.

A graph refers to an undirected weighted graph. Given a graph G, its set of vertices is $V(G)$, its set of edges is $E(G)$, and its weights are $W(G)$. An edge $(u, v) \in E(G)$ denotes an undirected link between vertices u and v so (u, v) is the same as (v, u). The weight of the edge (u, v) in G is $W_G(u, v)$; as only one graph is involved, we simply write $W(u, v)$ for convenience. Given a subset of vertices $X \subseteq V(G)$, let $G[X]$ denote the subgraph induced by X.

Definition 1 (Consensus tree) *Let $S = \{T_1, \ldots, T_k\}$ be a set of k phylogenetic trees on the same set of species $L = L(T_1) = \cdots = L(T_k)$. A* consensus tree *of S is a phylogenetic tree T_c with $L(T_c) = L$ that summarizes all of the trees in S.*

The main challenge considered in this paper is to find a consensus tree T_c of S that accurately represents the topology as well as the branch lengths of all the trees in S. In the case of building a consensus tree, the objective function (OF) of the method can be defined as follows:

$$OF = \sum_{i=1}^{k} dist(T_i, T_c), \tag{1}$$

where k is the number of input trees, T_c is the consensus tree, and $dist(T_i, T_c)$ is a distance metric between input phylogenetic tree i (denoted T_i) and T_c. The objective is to determine the consensus tree, T_c, that minimizes the sum of distances across all input trees, thereby optimizing the agreement between T_c and the input set. This optimization seeks to capture the central tendency of the phylogenetic relationships encoded within the input trees while minimizing discordance.

3 Related work

Given a set of phylogenetic trees, many methods exist for defining a consensus tree [11, 12]. The most well-known types of consensus trees are the strict consensus tree [13], the majority-rule consensus tree [14], the extended majority consensus tree (also referred to in the literature as the greedy consensus tree) [11, 12], and the frequency difference consensus tree [15] (called the plurality consensus tree in [16]). All consensus tree inference methods are based on the topology of the input phylogenetic trees. They mostly focus on the representation of each clade in the input trees.

Unfortunately, none of the methods described above is able to reconstruct branch lengths. Recently, a novel method for generating a consensus tree that incorporates branch lengths was proposed by Sifat and Tahiri [17]. The experimental analysis conducted was limited, involving the utilization of branch length and edge frequency to derive the MST. The obtained results were then compared with a majority-rule consensus tree, revealing a close similarity.

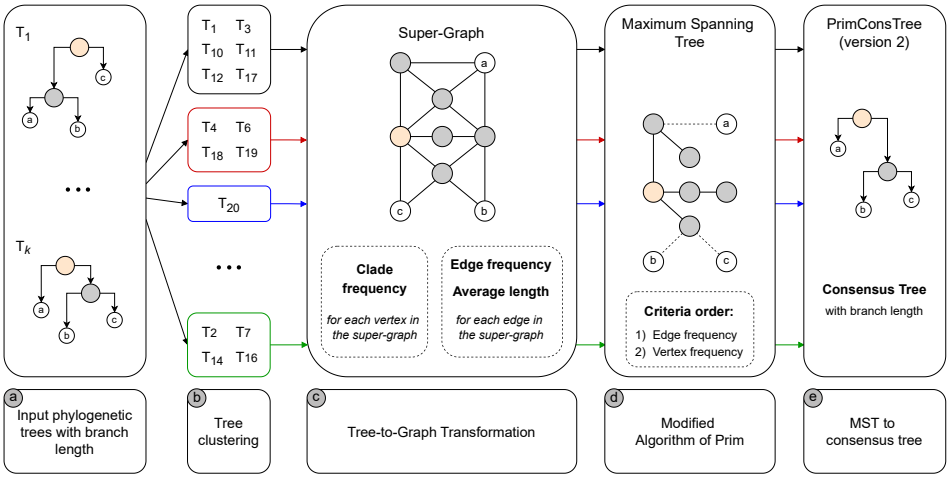

Figure 1: Comprehensive visualization of the system architecture overview, illustrating the three main steps of the approach (a-e). The process initiates with the initial input trees and concludes with the generation of one consensus tree per cluster, through the modified Prim algorithm [18]. From step (b), each color of the arrows represents an independent instance of the pipeline.

4 Method

In this study, we address the complex problem of integrating simultaneously edge frequency and branch length into the construction of a consensus tree. Our method pipeline, depicted in Figure 1, involves a series of steps to derive a consensus tree from the input phylogenetic trees. We subdivide the new method in two main phases, in addition to a preprocessing phase. In the preprocessing phase, we partition the input trees into several clusters (see Section 4.1), then each cluster can be processed independently by the following main phases. In the first phase, we transform a set of trees into a super-graph. In the second phase, we use an adapted version of the algorithm of Prim to construct a Maximum Spanning Tree (MST) of the super-graph. This MST serves as a base to create the consensus tree. We describe the first phase, and the second phase. Figure 1 (a,b) presents the clustering phase, Figure 1 (c) presents the first phase, and Figure 1 (d,e) presents the second phase of our method.

4.1 Clustering

Phylogenetic analysis involves three distinct steps. In the first step, researchers collect data, such as genomic, proteomic, and metabolomic data, for the different taxa under study (e.g., genes, species, morphology). The next step is to apply a tree reconstruction method to the collected data. Many of these methods produce several potential trees for the given data set. Often, hundreds or thousands of trees can be obtained. In the final step, a consensus tree is calculated from the candidate trees to reconcile conflicts, summarize information, and mitigate the large number of potential solutions in the evolutionary story. Although many consensus tree methods exist (see Section 3), they generally generate a single tree, which poses problems such as loss of information and susceptibility to outliers.

Given k genes defined across n species, the problem is to identify the optimal partition of phylogenetic trees that exhibit similar patterns of evolutionary history while accounting for outliers. This intermediate step in constructing a consensus tree facilitates the resolution

of conflicts among trees and emphasizes the potential for various alternative consensus trees [3, 4]. Figure 1 (b) illustrates the clustering step in the process. We have added the choice of k-medoids to create a homogeneous clustering with the Silhouette (SH) index [19]. We selected the k-medoids clustering method for its ability to handle variability in phylogenetic tree topologies and minimize the influence of outliers. This method optimizes the SH index [19] to form clusters with well-supported topological similarities. Its non-hierarchical nature avoids biases that may arise in hierarchical approaches, making it suitable for datasets with complex evolutionary patterns. The SH index is a method used to interpret and validate the consistency within clusters of data, providing a concise graphical representation of how well each object has been classified. This technique was introduced by [19]. The SH value measures how similar an object is to its cluster (cohesion) compared to other clusters (separation). This value ranges from -1 to +1, with a high value indicating that the object is well-matched to its cluster and poorly matched to neighboring clusters. If most objects have high SH values, the clustering configuration is considered appropriate. Conversely, if many points have low or negative values, the clustering configuration may have too many or too few clusters.

All subsequent steps following clustering will independently process the different clusters obtained by k-medoids (see Figure 1, which shows the various colored arrows). From now on, the set of input trees S will refer to the set of trees within a given cluster.

4.2 Tree-to-graph transformation

In this phase, depicted in Figure 1 (c), we present a dynamic tree-to-graph transformation, enabling the conversion of a set of input phylogenetic trees into an undirected weighted graph called the super-graph. We explain the dynamic naming of internal nodes followed by the construction of the super-graph. During this phase, we also detail the computation of edge frequency, clade frequency, and average edge lengths.

Definition 2 (Dynamic tree) *A dynamic tree T is a tree structure that satisfies the properties of a phylogenetic tree. Additionally, every internal node $u \in I_T$ is named according to the process described below.*

The super-graph construction brings together nodes and edges from multiple trees. To identify the occurrences of the same internal node in distinct trees, we transform each phylogenetic tree of S into a dynamic tree by identifying its internal nodes. Each internal node is named after its corresponding clade, such as two nodes $u \in I(T_i)$ and $v \in I(T_j)$ are considered equivalent if $C_u = C_v$. Therefore, the same node occurring in multiple trees can reconcile different branching patterns. For example, consider how internal nodes are labeled in Figure 2 (a). The node *abc* appears in two different trees, each time with a distinct underlying topology. It is worth noting that all trees in S share the same set of leaves, thus leaving every root node with the same name after the name attribution of the internal nodes.

It follows that after this step, edges are also identified according to the new names of internal nodes. Two edges $(u, v) \in E(T)$ and $(u', v') \in E(T')$ are equivalent if $u = u'$ and $v = v'$. We do not consider edge lengths or edge directions since the graph is undirected. By opposition, the edges are distinct if $u \neq u'$ or $v \neq v'$.

Definition 3 (Dynamic trees union) *Given a set of dynamic trees D, the dynamic tree union operation is formally noted $\bigcup_{T_i \in D} T_i$. The result of the operation is an undirected unweighted graph G; the vertices of $V(G)$ are split in two disjoint subsets such as $V(G) = I(G) \cup L(G)$, where $I(G)$ denote internal nodes from the trees and $L(G)$ denote leaf nodes from the trees; for clarification we always use the terms internal / leaf vertices for a graph and internal / leaf*

nodes for a tree. Vertices are given by $I(G) = \bigcup_{T_i \in D} I(T_i)$ *and* $L(G) = \bigcup_{T_i \in D} L(T_i)$*, and edges are given by* $E(G) = \bigcup_{T_i \in D} E(T_i)$*.*

The super-graph $\mathcal{G}$ is the result of the dynamic tree union operation applied on the set of input trees S. Every tree shares the same root, so we call the unique vertex corresponding to the root in $V(\mathcal{G})$ the *root vertex* for convenience. Additionally, an *edge frequency* is attached to each edge $(u, v) \in E(\mathcal{G})$ as the edge weight $W(u, v)$. Given an edge (u, v), its frequency $W(u, v)$ is the number of input trees that contain this edge. Formally, let $S_{(u,v)} = \{T \in S : (u, v) \in E(T)\}$ denote the set of trees that contain the edge (u, v) the formula for edge frequency is defined as follows:

$$W(u, v) = |S_{(u,v)}|. \tag{2}$$

Similarly, a *clade frequency* is assigned to each internal vertex in the super-graph. Let u be an internal vertex in $I(\mathcal{G})$, we note $F(u)$ the clade frequency associated u and give the following formal definition. Given $S_u = \{T \in S : u \in V(T)\}$ the set of trees support the clade C_u, and r the root vertex, the formula for clade frequency is defined as follows:

$$F(u) = \begin{cases} |S_u|, & \text{if } u \neq r. \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Clade frequency provides crucial insight into how often a clade is supported in the input trees. The clades of a phylogenetic tree are highly significant because, collectively, they describe the entire topology of the tree. Consequently, clade frequency constitutes an essential metric in the construction of consensus trees, providing an objective measure for representing the recurring topologies across the input trees.

Originally, the clade frequency of the root vertex should be equal to $|S|$ because the corresponding node is present in all trees. We deliberately put it to 0 to avoid a biased situation where the root vertex gains undue importance merely due to its presence in every tree. This approach allows for a more balanced and accurate representation of clade significance across the super-graph.

Edge and clade frequency are closely related to each other but regardless, each gives important insight on the topology of the input trees. While the presence of an internal node u in a tree relates to the presence of a clade, it does not provide details on the topology of the underlying subtree $T[u]$. On the other hand, because every internal node is named, each edge depicts an explicit link between two nodes, which is much more accurate and therefore, should be given more attention. However, clade frequency is still very important when tree topologies are too conflicted to refer to explicit edges. For instance, in Figure 2 (a), leaf a is never connected with the same edge; nevertheless, the clade abc is represented in two trees out of three. Considering only edge frequency, a might be connected to the root in Figure 2 (d), leaving clade abc unresolved in Figure 2 (e). The clade frequency suggests a preference for the edge (abc, a) as it contributes to including the clade abc.

The last attribute to be attached to the super-graph is *average edge length*. It describes the average length of a given edge across every tree of S that includes the edge. Other trees are omitted because they do not hold relevant information about the length of the edge. The formula to compute average edge length is defined as follows:

$$D(u, v) = \frac{\sum_{T \in S_{(u,v)}} dist_T(u, v)}{W(u, v)}, \tag{4}$$

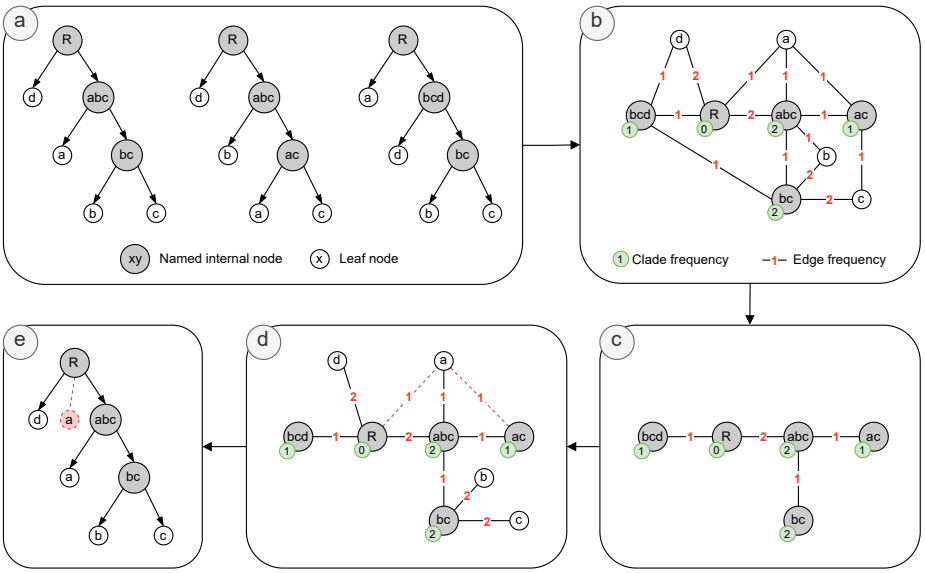

Figure 2: This small example outlines how edge and clade frequencies were utilized to retain topological information in PrimConsTree. (a) Three input trees, with already named internal nodes (with root *abcd* as R); (b) Construct a super-graph encompassing all nodes, incorporating edge and clade frequencies; (c) Derive the Maximum Spanning Tree (MST) from the super-graph, with a focus on internal vertices; (d) Establish connections between leaves and internal vertices based on edge and clade frequencies; (e) The consensus tree is obtained by PrimConsTree by removing unnecessary internal nodes from the MST. In (d, e), red edges outline how ignoring clade frequency could lead to the loss of an important information on the clade *abc*.

where $S_{(u,v)} \subseteq S$ is the subset of trees containing an edge between u and v, such as $S_{(u,v)} = \{T \in S | (u,v) \in E_T\}$, $dist_T(u,v)$ is the length of this edge in T and $W(u,v)$ is the edge frequency of the (u,v) from Equation 2.

In the context of our analysis, the potential impact of outliers is minimized by the clustering. Therefore, every tree should have its importance in the result and the average is the most appropriate metric because it takes every tree into account.

Lemma 1 (Connectivity of the super-graph) *Let $\mathcal{G}$ be the super-graph resulting from the dynamic trees union operation, applied on the set of input trees S. The graph $\mathcal{G}$ is connected and the subgraph $\mathcal{G}[I(\mathcal{G})]$ restricted to internal vertices is also connected.*

Figure 2 describes the process used to build the super-graph. Initially, the super-graph $\mathcal{G}$ is empty, first, we include the set of leaves such that $L(\mathcal{G}) = L(S)$, and then, we sequentially incorporate information from each tree $T \in S$. When adding a tree T, the graph internal vertices are updated as $I(\mathcal{G}) = I(\mathcal{G}) \cup I(T)$ and the graph edges are updated as $E(\mathcal{G}) = E(\mathcal{G}) \cup E(T)$. For each edge $(u,v) \in E(T)$, the edge frequency is increased by one and the average length is incremented by $dist_T(u,v)$. The clade frequency $F(u)$ of each node $u \in T$ is also increased by one. Finally, the average length of each edge is divided by its frequency.

At the end of the first phase, the main properties of the input trees are retained in the super-graph: 1) Tree topologies are retained firstly in the graph topology and secondly in its attributes, including edge frequencies W and clade frequencies F and 2) Branch length data is retained in the average edge length attribute D.

4.3 Consensus tree construction from the super-graph

In the following phase, depicted in Figure 1(d,e), the analysis involves the super-graph $\mathcal{G}$, including edge and clade frequency and average edge length. Figure 1d builds a specific Maximum Spanning Tree (MST) of the super-graph and uses the MST as a base to generate a consensus tree T_c. Indeed, an MST is a straightforward and efficient way to find a tree structure that maximizes the values of its attributes (i.e., edge and clade frequency), consequently maximizing the accordance of its topology with the input trees. Another advantage of the MST is that it directly takes edges from the super-graph, which already retains branch length data.

In the context of a consensus tree, the resulting consensus tree must have the same set of leaves as the phylogenetic trees used as input. To ensure that the MST does not compromise this condition, we divide its construction into two main steps. The first step utilizes the subgraph $\mathcal{G}[I(\mathcal{G})]$ restricted to internal vertices only, and yields its Maximum Spanning Tree (MST). Indeed, constructing the MST from all vertices could lead to the conversion of leaf vertices into internal nodes within the MST, a scenario deemed undesirable. In the second step, the vertices in $L(\mathcal{G})$ are connected to the MST using the function AttachLeaves.

The process of Figure 1e follows the course of the original algorithm of Prim. It starts by including the root vertex in the MST. Then, at each iteration, it looks for every edge that has exactly one endpoint included in the MST and includes the one of higher weight. Subsequently, the process repeats until all vertices are included in the MST. The particularities of our modified Algorithm of Prim lie in two points. Firstly, the set of leaf vertices $L(\mathcal{G})$ is connected independently after the rest of the vertices. Secondly, in addition to the weight (i.e., edge frequency), the clade frequency is used to choose the optimal edge.

Let (u, v) be a candidate edge at a given iteration, where $u \in I(\mathcal{G})$ is included in the MST, $v \in I(\mathcal{G})$ is not included in the MST and $(u, v) \in E(\mathcal{G})$. To choose the optimal edge, Figure 2b first finds the edge of highest edge frequency $W(u, v)$. Indeed, edges give the most precise insights into the topology so edge frequency must be considered *in priority*.

In case multiple candidates of maximal edge frequency are available, clade frequencies of the edge endpoints are taken into account. First, the algorithm considers the clade frequency of the vertex that is not included in the MST, in other words, it maximizes $F(v)$. Then, if multiple optimal edges are still available, the clade frequency of the vertex included in the MST, $F(u)$, is maximized.

Then, Figure 2d is used to connect each leaf vertex to the most suitable internal node. This is done in a very similar way to the rest of the MST. Given an internal vertex $u \in I(\mathcal{G})$ and a leaf vertex $v \in L(\mathcal{G})$ the criteria are maximized in the following order: first edge frequency $W(u, v)$, and then clade frequency of the internal vertex $F(u)$.

The final step is to use the MST as a base to create a valid consensus tree T_c that respects the following conditions. Firstly, its edges must be directed and must have edge length attached. Secondly, every internal node must have at least two children. Finally, the set of leaves must be equal to the initial set of leaves such that $L(T_c) = L(S)$.

The consensus tree is directed by positioning the root as the root vertex. For each edge (u, v) in the MST, the edge is added $E(T_c)$, pointing away from the root. At the same time, the length $dist_{T_c}(u, v) = D(u, v)$ is attached to the edge.

Finally, we define two types of extra nodes that must be removed in order to get a proper consensus tree: *unnecessary* internal nodes and *redundant* internal nodes. Any internal node $u \in I(\mathcal{G})$ that became a leaf in the MST, is considered unnecessary and can easily be removed because it does not have children. On the other hand, an internal node is redundant if it has only one child (leaf or internal node). Let u be a redundant node, $p(u)$ is its parent, and $c(u)$ is its unique child. Then the node u is deleted and a new edge $(p(u), c(u))$ is added. The length of the new edge is the total length of the previous path such that $dist_{T_c}(p(u), c(u)) = dist_{T_c}(p(u), u) + dist_{T_c}(u, c(u))$. At the end of this phase, a proper consensus tree T_c is returned.

We would like to emphasize that the output of PrimConsTree version 2 is not necessarily a binary tree. While the recursive edge selection process might initially suggest a binary structure, the inclusion of non-binary relationships remains possible. Specifically, during the node removal step, internal nodes with multiple children that do not contribute to distinct clades can lead to the emergence of non-binary nodes in the final consensus tree. The node removal process ensures that non-binary structures can be maintained when internal nodes with multiple children fail to contribute uniquely to the topology. This step allows for flexibility, supporting non-binary consensus trees when appropriate, reflecting ambiguous or weakly supported evolutionary relationships. The AttachLeaves plays a critical role in handling terminal edges for taxa represented by single nodes, ensuring that each taxon is properly attached to the consensus tree. This step is essential in maintaining both binary and non-binary relationships, especially when dealing with taxa that form terminal clades.

5 Conclusions

In this paper, we extended PrimConsTree version 1, a graph-based approach for constructing consensus trees with balanced branch lengths. To achieve this, we clustered the input trees and proposed new key criteria, such as clade frequency, to derive the maximum spanning tree. We provide a detailed description of the supergraph construction and an enhanced version of the well-established Prim algorithm.

For future research directions, we propose exploring the extension of the applicability of PrimConsTree version 2 to address the supertree problem, particularly in situations where the number of leaves in gene trees may vary. Furthermore, there is potential for additional refinements to enhance the ability of the algorithm to generate a consensus tree that closely aligns with the gene trees. Experimental results related to a study of the evolution of Archaebacteria and the simulation analysis will be reported in the full version of this paper. These results will include reconstructions of evolutionary scenarios, with a focus on the detection of horizontal gene transfer and recombination events. The simulation analysis will evaluate the accuracy and performance of PrimConsTree version 2 under varying conditions, such as differences in the number of leaves and tree topologies. A detailed comparison with other methods will also be included to examine the characteristics and potential challenges of the algorithm. These findings will provide insights into the evolutionary processes underlying the history of Archaebacteria. Due to space constraints, the proof of the theorem will also be provided in the full version of this paper.

6 Funding

This work was supported by the Natural Sciences and Engineering Research Council of Canada, Fonds de recherche du Québec - Nature and Technologies, and the University of Sherbrooke grant.

References

[1] J.E. O'Reilly, P.C. Donoghue, The efficacy of consensus tree methods for summarizing phylogenetic relationships from a posterior sample of trees estimated from morphological data, Systematic biology **67**, 354 (2018).

[2] J.H. Degnan, M. DeGiorgio, D. Bryant, N.A. Rosenberg, Properties of consensus methods for inferring species trees from gene trees, Systematic Biology **58**, 35 (2009).

[3] N. Tahiri, B. Fichet, V. Makarenkov, Building alternative consensus trees and supertrees using k-means and robinson and foulds distance, Bioinformatics **38**, 3367 (2022).

[4] N. Tahiri, M. Willems, V. Makarenkov, A new fast method for inferring multiple consensus trees using k-medoids, BMC evolutionary biology **18**, 1 (2018).

[5] J. Felsenstein, Phylogenies and the comparative method, The American Naturalist **125**, 1 (1985).

[6] M.W. Hahn, T. De Bie, J.E. Stajich, C. Nguyen, N. Cristianini, Estimating the tempo and mode of gene family evolution from comparative genomic data, Genome research **15**, 1153 (2005).

[7] S.L. Kosakovsky Pond, S.D. Frost, Not so different after all: a comparison of methods for detecting amino acid sites under selection, Molecular biology and evolution **22**, 1208 (2005).

[8] E.M. Volz, K. Koelle, T. Bedford, Viral phylodynamics, PLoS computational biology **9**, e1002947 (2013).

[9] V. Lefort, R. Desper, O. Gascuel, Fastme 2.0: a comprehensive, accurate, and fast distance-based phylogeny inference program, Molecular biology and evolution **32**, 2798 (2015).

[10] B. Rannala, The art and science of species delimitation, Current Zoology **61**, 846 (2015).

[11] D. Bryant, A classification of consensus methods for phylogenetics, DIMACS series in discrete mathematics and theoretical computer science **61**, 163 (2003).

[12] J. Jansson, C. Shen, W.K. Sung, Improved algorithms for constructing consensus trees, Journal of the ACM (JACM) **63**, 1 (2016).

[13] M. Wilkinson, J.L. Thorley, Efficiency of strict consensus trees, Systematic Biology **50**, 610 (2001).

[14] M. Wilkinson, Majority-rule reduced consensus trees and their use in bootstrapping., Molecular Biology and evolution **13**, 437 (1996).

[15] J. Jansson, W.K. Sung, S.A. Tabatabaee, Y. Yang, A Faster Algorithm for Constructing the Frequency Difference Consensus Tree, in *41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024)* (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024), Vol. 289 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 43:1–43:17

[16] J.D. Velasco, The foundations of concordance views of phylogeny, Philosophy, Theory, and Practice in Biology **11** (2019).

[17] M.H.R. Sifat, N. Tahiri, A new algorithm for building comprehensive consensus tree, in *Graphs and more Complex structures for Learning and Reasoning: Proceedings of the 38th Annual AAAI Conference on Artificial Intelligence* (2024)

[18] V. Jarník, O jistém problému minimálním, Práca Moravské Prírodovedecké Spolecnosti **6**, 57 (1930).

[19] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of computational and applied mathematics **20**, 53 (1987).